Adopting new technologies in the LHCb Gauss simulation framework

Dominik Müller on behalf of the LHCb collaboration

CERN

CHEP 2018, Sofia

9th of July 2018

Motivation

- LHC Run 3: LHCb large increase in luminosity
  - Very challenging for computing
  - Modernize all LHCb software
  - CERN-LHCC-2018-007
- Necessary to update the simulation framework!
  - Multithreaded (Gaudi and Geant4)
  - New framework built on:

Gaussino

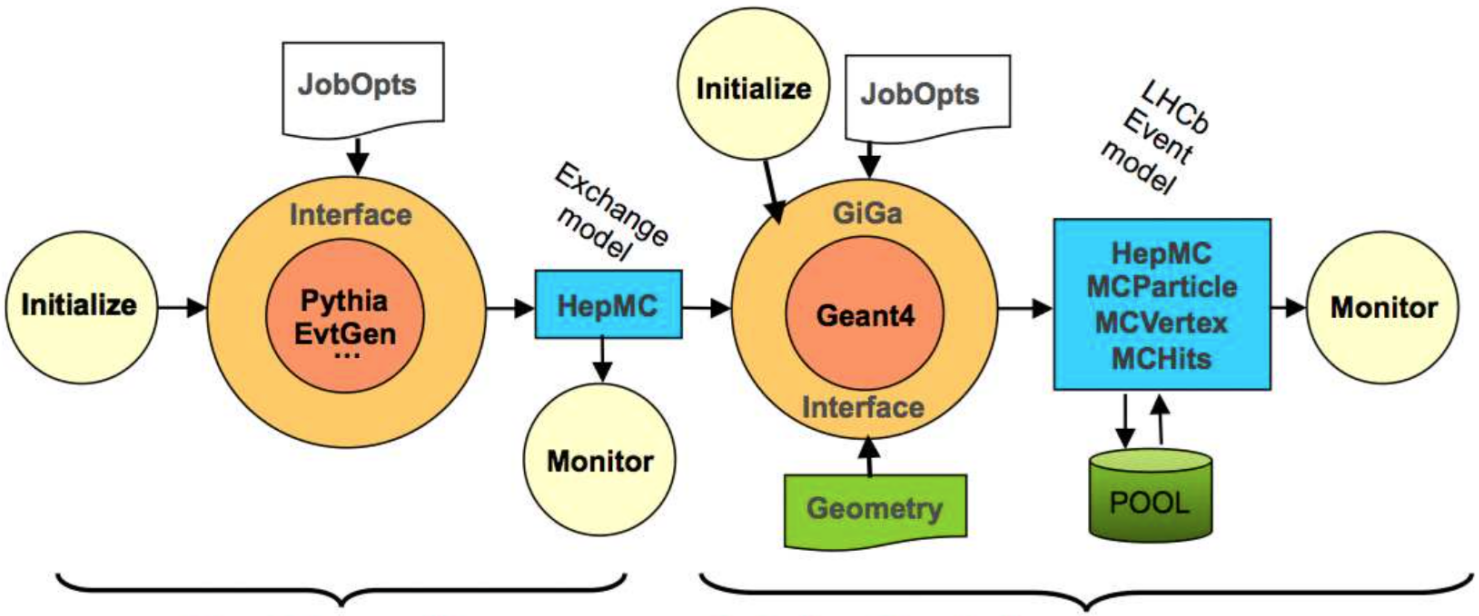
- LHCb independent core framework
- Aiming at a wider audience
- Improved maintainability

Gauss

The LHCb simulation framework

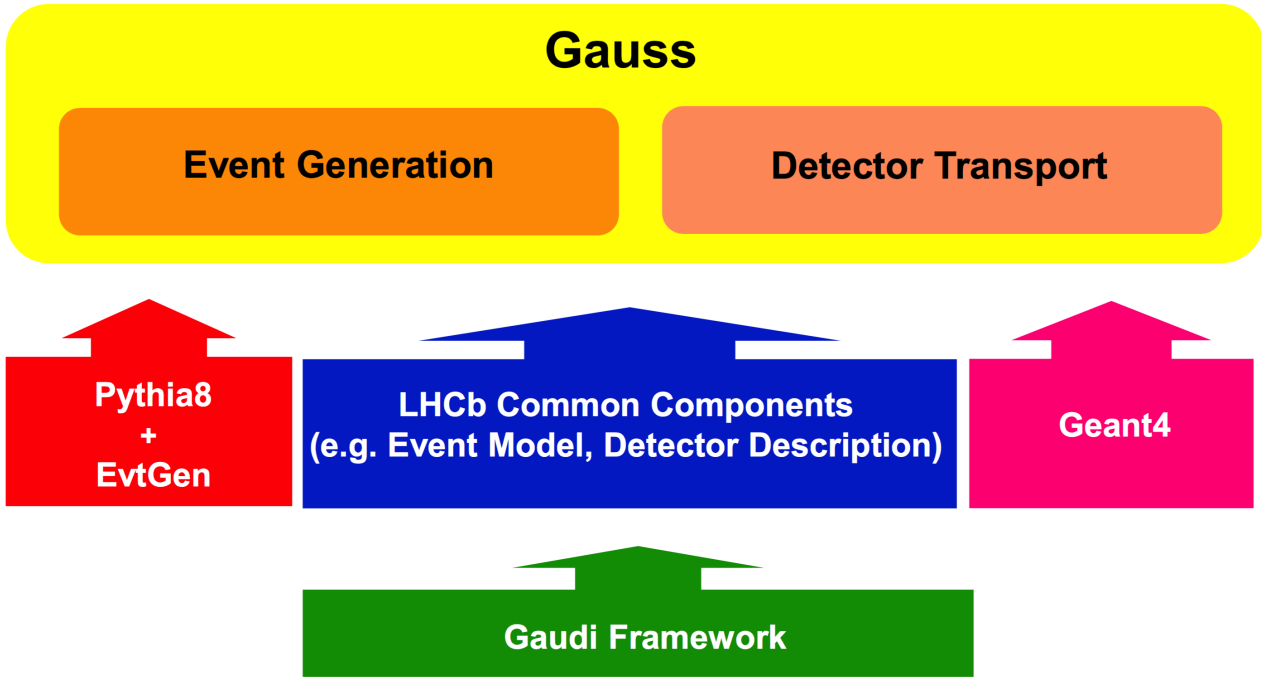
Current Gauss – A quick overview M. Clemencic et al. 2011 J. Phys. Conf. Ser. 331 032023

- Flexible framework to combine and unify various generators.



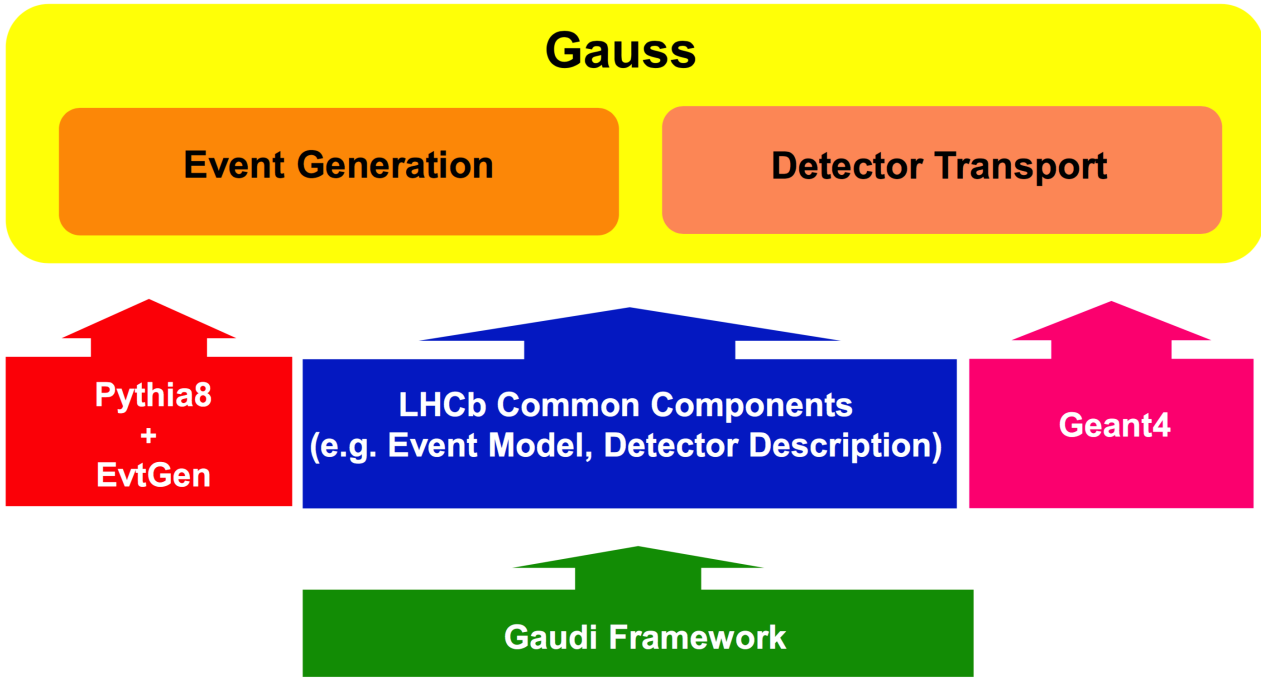
Usage in LHCb

- Efficiencies and shapes
- Usually per measurement: **modular** generation phase
- Typically: Pythia8 minbias → EvtGen decays → Geant4 simulation



Shortcomings

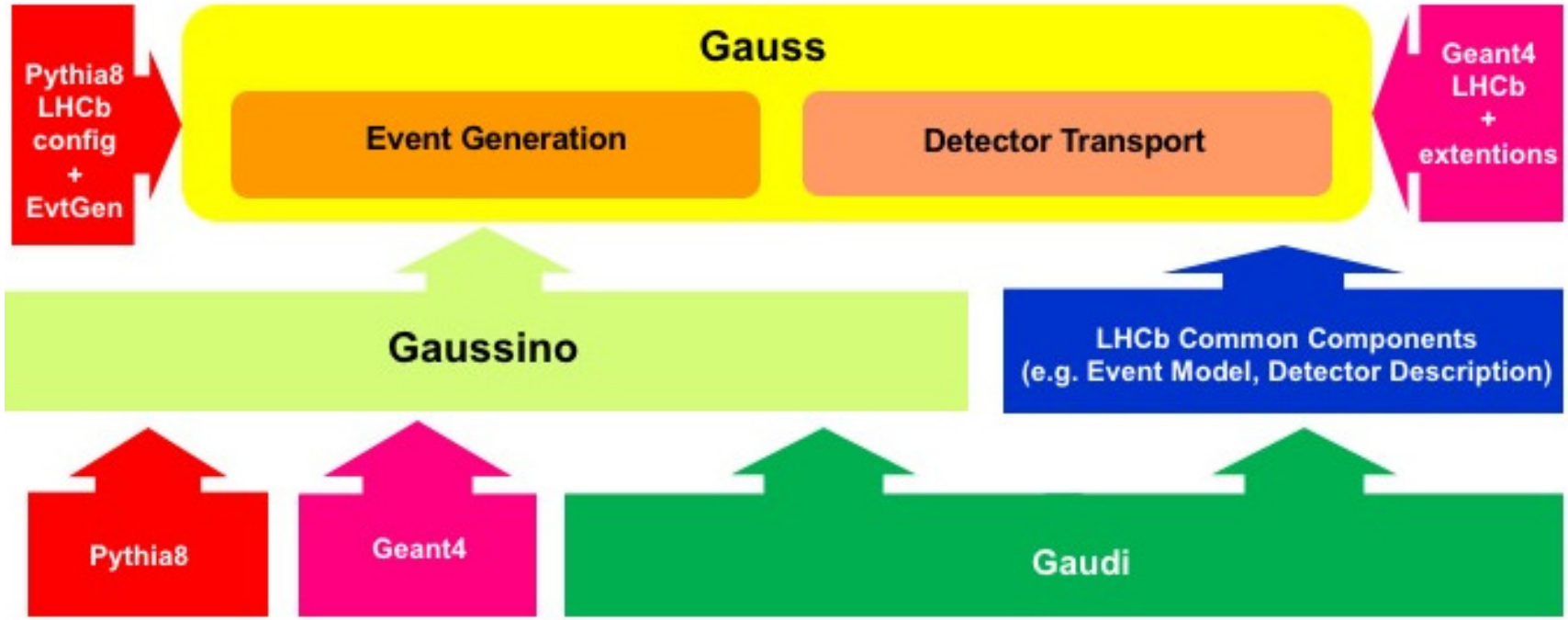
- Single-threaded:
  - limits use of memory limited resources
  - e.g. LHCb trigger farm during LHC downtime
- Simulation phase designed around Geant4
- Framework of running experiment:
  - Has grown a lot, redundant code, ...



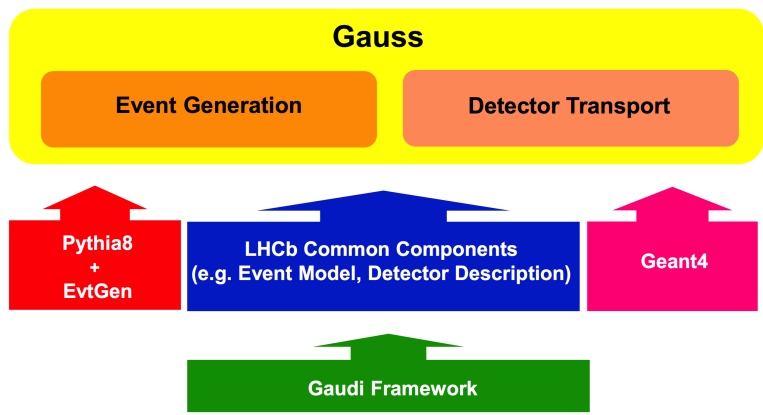
Shortcomings

- Single-threaded:
  - limits use of memory limited resources
  - e.g. LHCb trigger farm during LHC downtime
- Simulation phase designed around Geant4
- Framework of running experiment:
  - Has grown a lot, redundant code, ...

The future of Gauss



Current framework



The plan in a nutshell

- **Multithreaded Gaudi functional**
- **Geant4 MT**
- Improved maintainability

Gaussino

- Provide generation and simulation interface.
- Internal data: HepMC3
- A complete simulation framework:
  - Define an interface to Generators (with Pythia8).
  - Interface to Geant (especially MT), but be flexible.
- Ideal test-bed for new developments

- Make use of the opportunity and clean up and modernise the code!

The plan in a nutshell

- **Multithreaded Gaudi functional**
- **Geant4 MT**
- Improved maintainability

Gaussino

- Provide generation and simulation interface.
- Internal data: HepMC3
- A complete simulation framework:
  - Define an interface to Generators (with Pythia8).
  - Interface to Geant (especially MT), but be flexible.
- Ideal test-bed for new developments

- Make use of the opportunity and clean up and modernise the code!

Gaussino

The generation phase

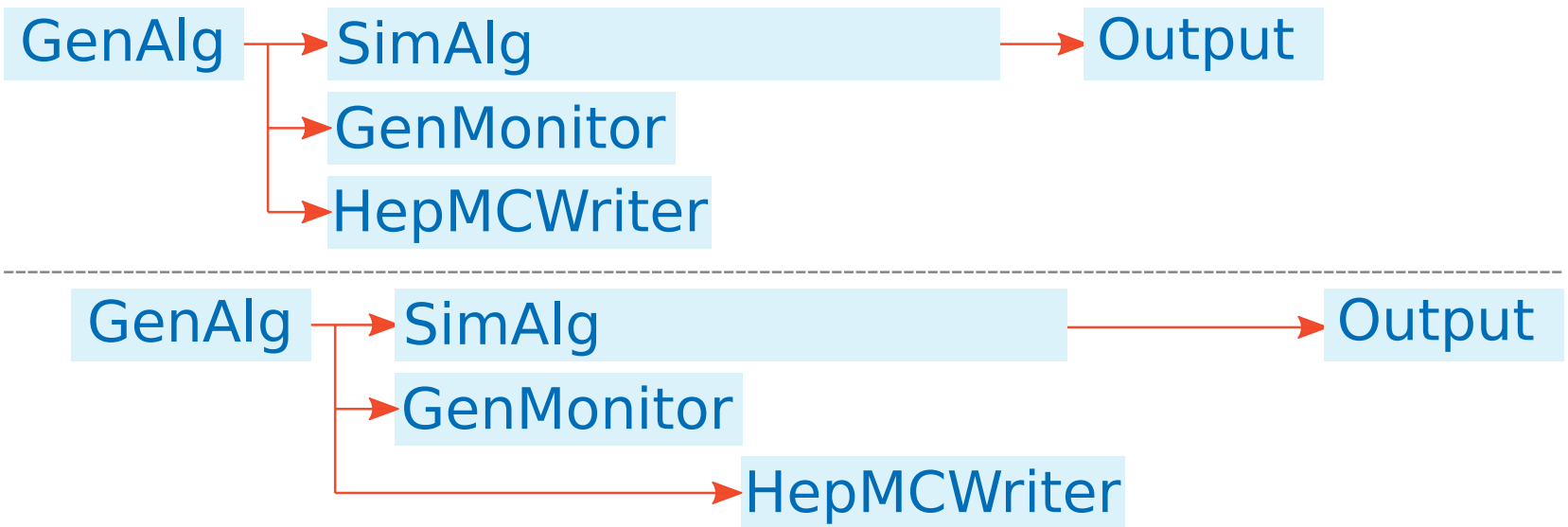
Gaudi functional: task-based parallism

- Each algorithm represents a 'task'

Algorithms

- Declare their data dependence in advance
- One instance called concurrently by all threads:

```
/*output data*/ operator()(/*input data*/) const
```



- Skeleton for Gaussino finalized
- Conversion to Gaudi functional **complete**
- Can generate events reproducible [with multiple threads]

External generators

- Shared between threads
- Unified interface
- Cannot control thread-safety of generators
  - Lock tool
  - Pythia8: investigating thread-local instances

HepMC3 migration

- Motivation: easier and thread-safe
- **We are the first to try this**
- **Complete**
- In communication with authors about feedback

- Skeleton for Gaussino finalized
- Conversion to Gaudi functional **complete**
- Can generate events reproducible [with multiple threads]

External generators

- Shared between threads
- Unified interface
- Cannot control thread-safety of generators
  - Lock tool
  - Pythia8: investigating thread-local instances

HepMC3 migration

- Motivation: easier and thread-safe
- **We are the first to try this**
- **Complete**
- In communication with authors about feedback

- Skeleton for Gaussino finalized
- Conversion to Gaudi functional **complete**
- Can generate events reproducible [with multiple threads]

External generators

- Shared between threads
- Unified interface
- Cannot control thread-safety of generators
  - Lock tool
  - Pythia8: investigating thread-local instances

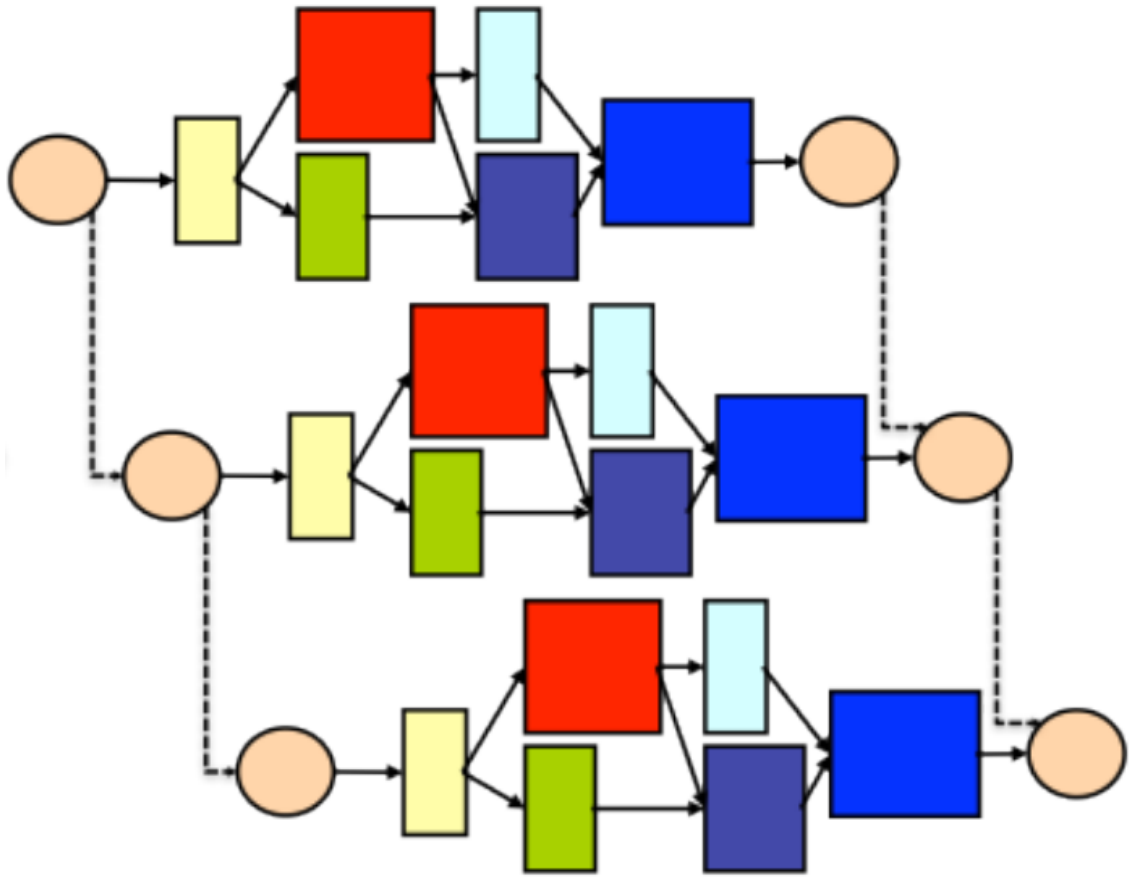
HepMC3 migration

- Motivation: easier and thread-safe
- **We are the first to try this**
- **Complete**
- In communication with authors about feedback

Random numbers

We ♡ reproducibility

'Random engine: a global singleton' and



won't work . . .

Prototype for a solution

- Largest predictable unit: algorithm's execution
- Create engine on the stack
- Pass it around as reference

```
... Generation::operator()( ... ) const {
  auto engine = createRndmEngine();
  ThreadLocalEngine::Guard rnd_guard(engine);
  nPileUp = m_pileUpTool->numberOfPileUp(engine);
  return; // Engine automatically destroyed
}
```

- Alternative: Thread-local global only valid in algorithm.

```
... Generation::operator()( ... ) const {
  auto engine = createRndmEngine();
  ThreadLocalEngine::Guard rnd_guard(engine);
  doSomething(); // Use engine via ThreadLocalEngine::Get()
  return; // Engine automatically destroyed and thread-local
  ↪  global invalidated
}
```

Prototype for a solution

- Largest predictable unit: algorithm's execution
- Create engine on the stack
- Pass it around as reference

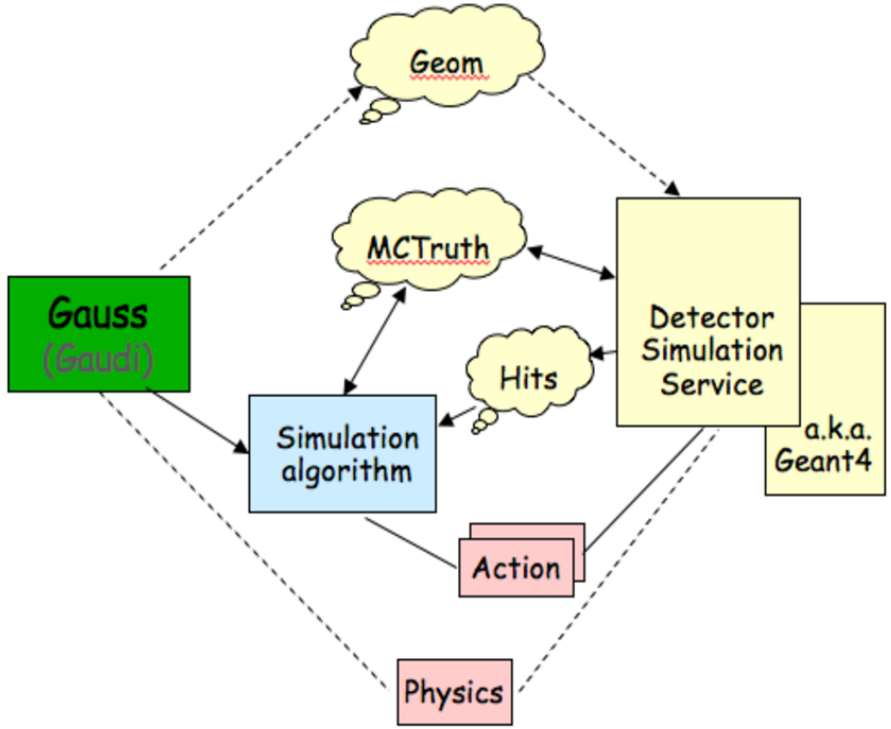
```
... Generation::operator()( ... ) const {
  auto engine = createRndmEngine();
  ThreadLocalEngine::Guard rnd_guard(engine);
  nPileUp = m_pileUpTool->numberOfPileUp(engine);
  return; // Engine automatically destroyed
}
```

- Alternative: Thread-local global only valid in algorithm.

```
... Generation::operator()( ... ) const {
  auto engine = createRndmEngine();
  ThreadLocalEngine::Guard rnd_guard(engine);
  doSomething(); // Use engine via ThreadLocalEngine::Get()
  return; // Engine automatically destroyed and thread-local
   ↪  global invalidated
}
```

Gaussino

The simulation phase

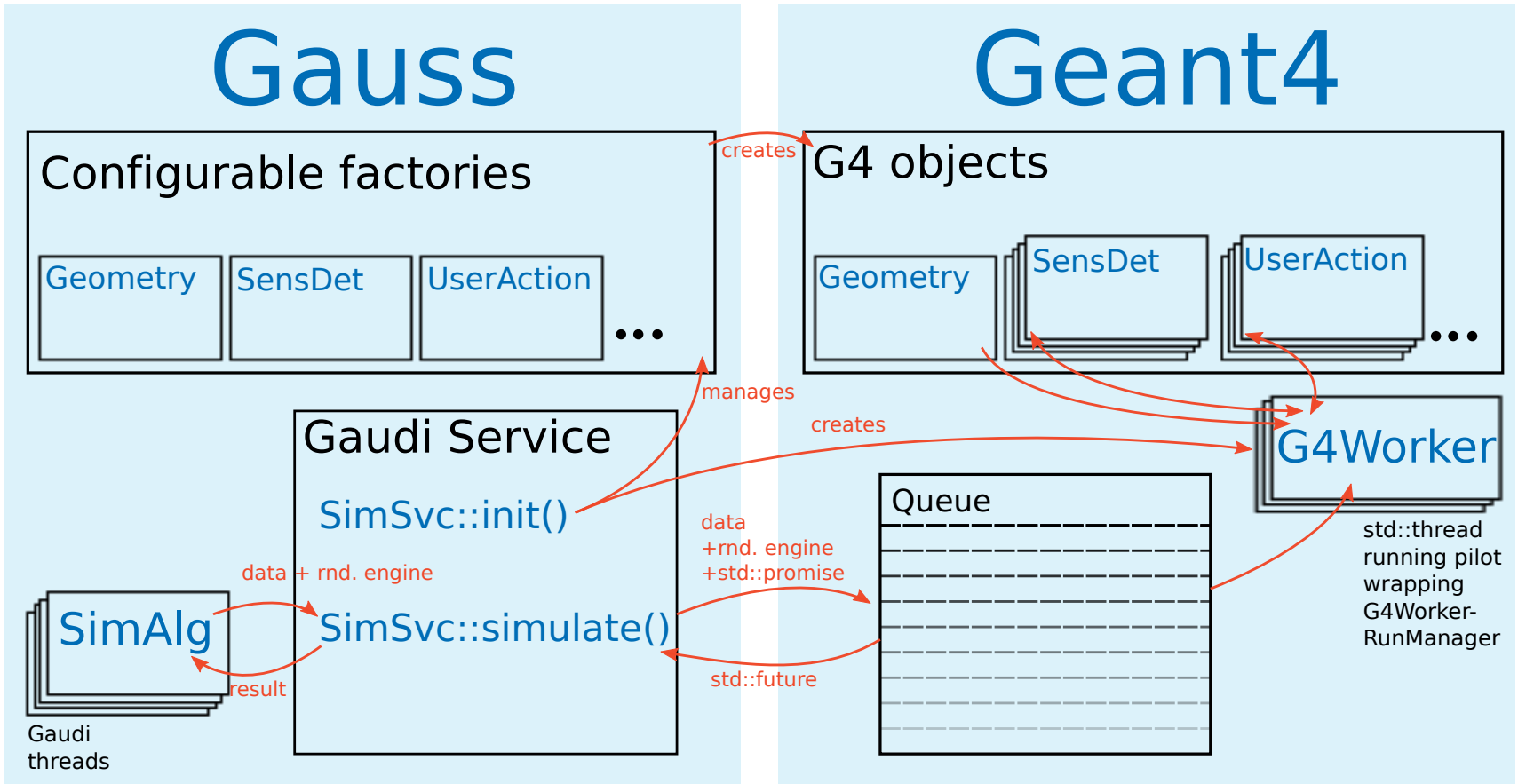


Simulation step

- Take inspiration from the generation: modular!
- A simulation service managing different backends
- Enable flexible configuration, e.g. **fast** simulation settings for
  - Pile-up – Spillover – Main event
  - Signal – other particles
  - See B. Siddi and M. Rama this morning

Interface to Geant4MT

- Separate Gaudi/Gauss from Geant4 as much as possible
- Gaudi tools as factories for G4 objects
  - All G4 objects managed by G4!
- Run G4 workers in individual threads
- **Flexible** assignment of simulation payloads



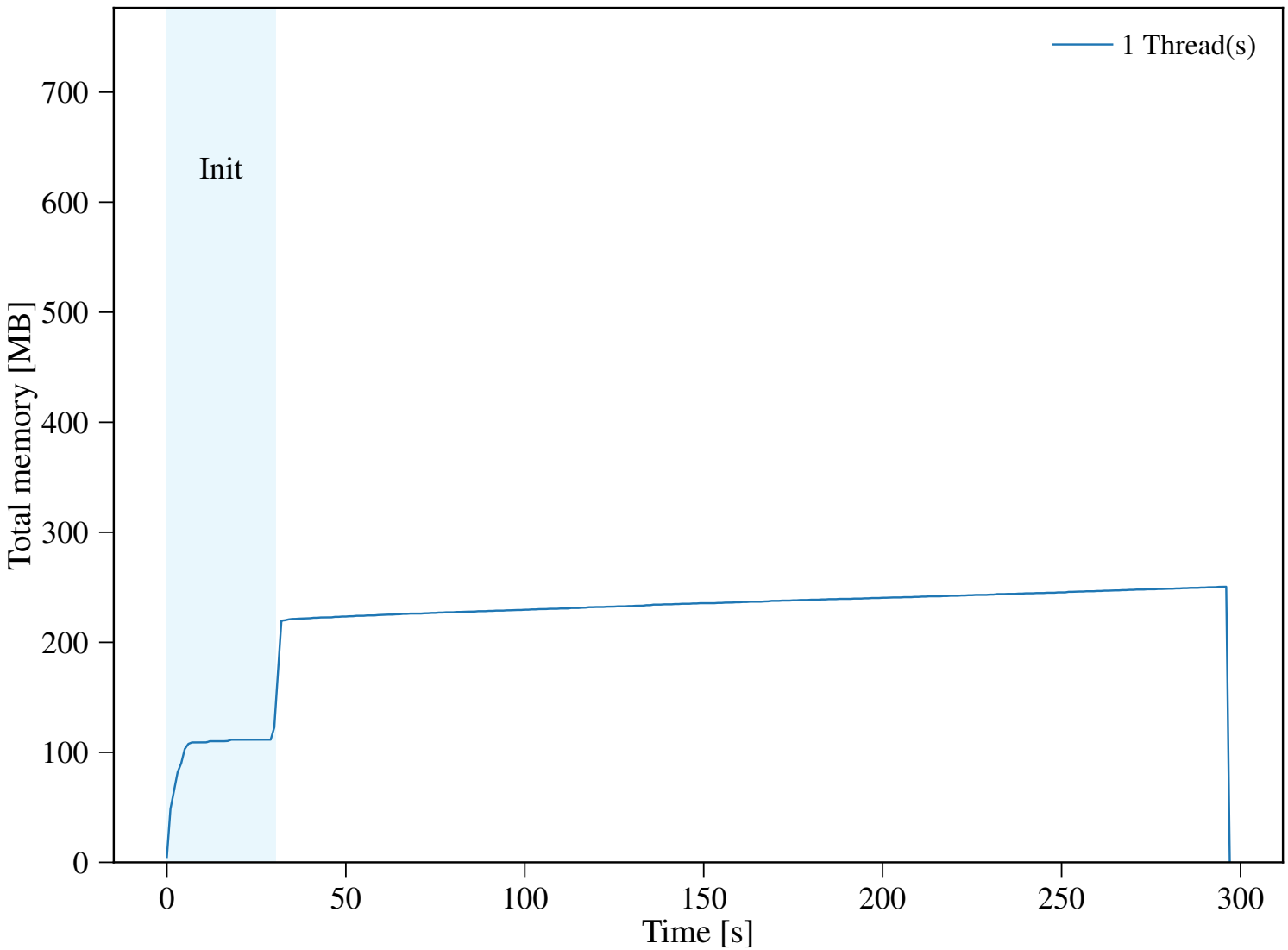
Implemented prototype

- Implemented prototype in Gaussino:
  - Initialization of the threads
  - Communication between Gaudi and Geant4 threads

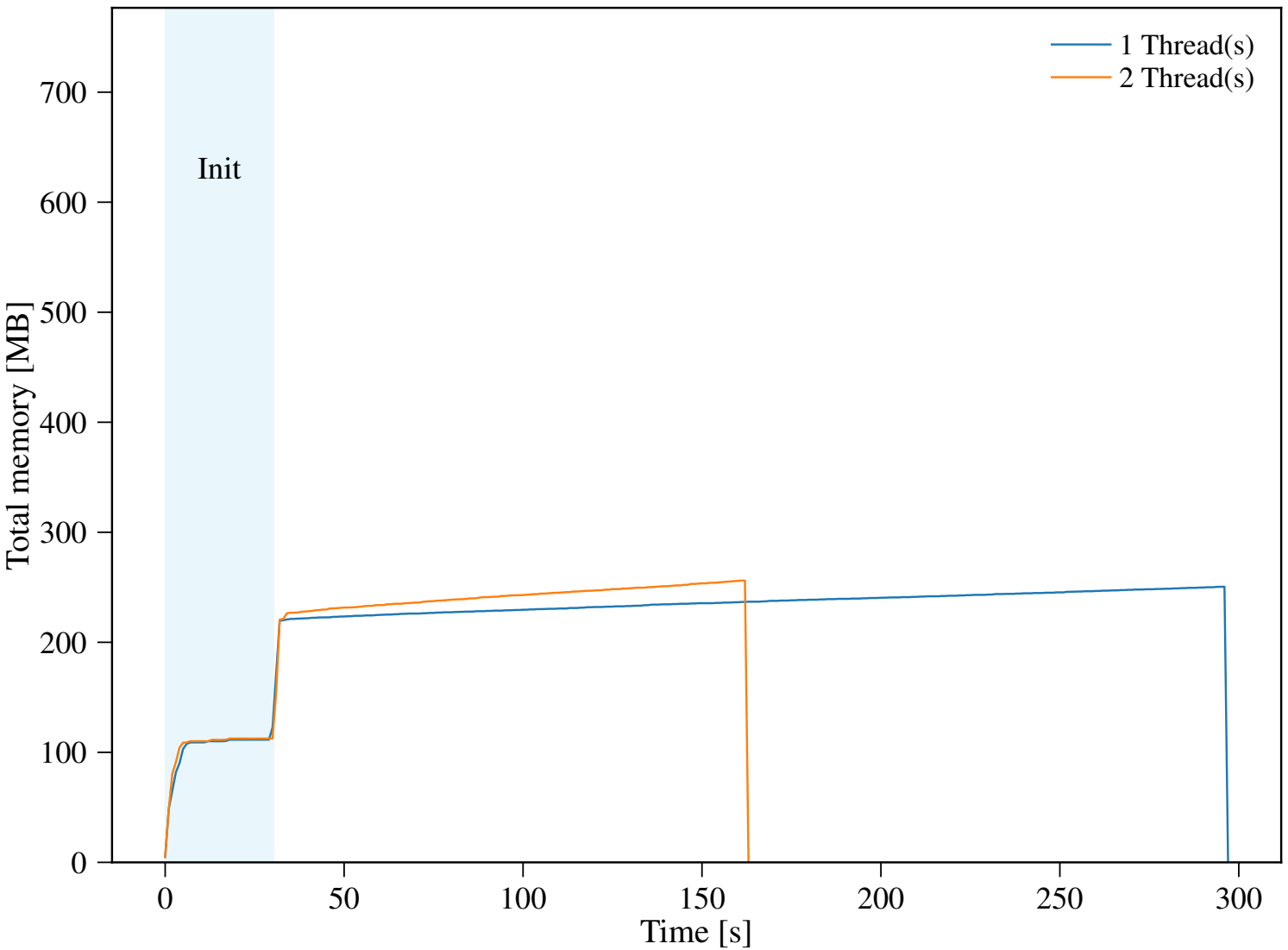
Tests

- Simulated $2 \times 2 \times 2\,\mathrm{m}^3$ iron cube
- Minimum bias events from Pythia8 (~ 10 pp interactions per event).
- Nothing returned from G4 yet
- Spawn 1 G4 thread for each Gaudi thread
- Performed on 2× Xeon E5-2630 v4 (20 cores + 20 HT)

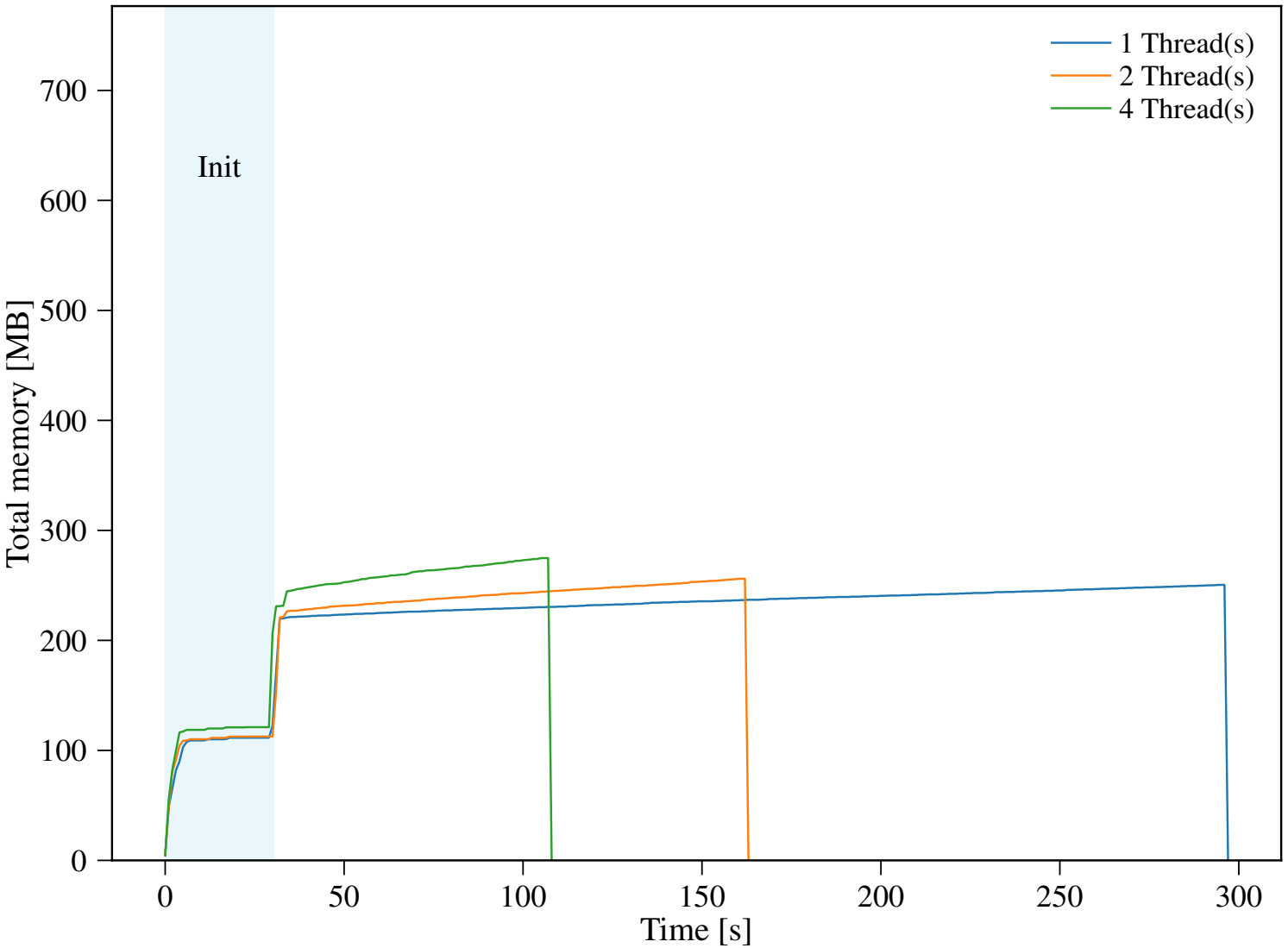
Memory scaling



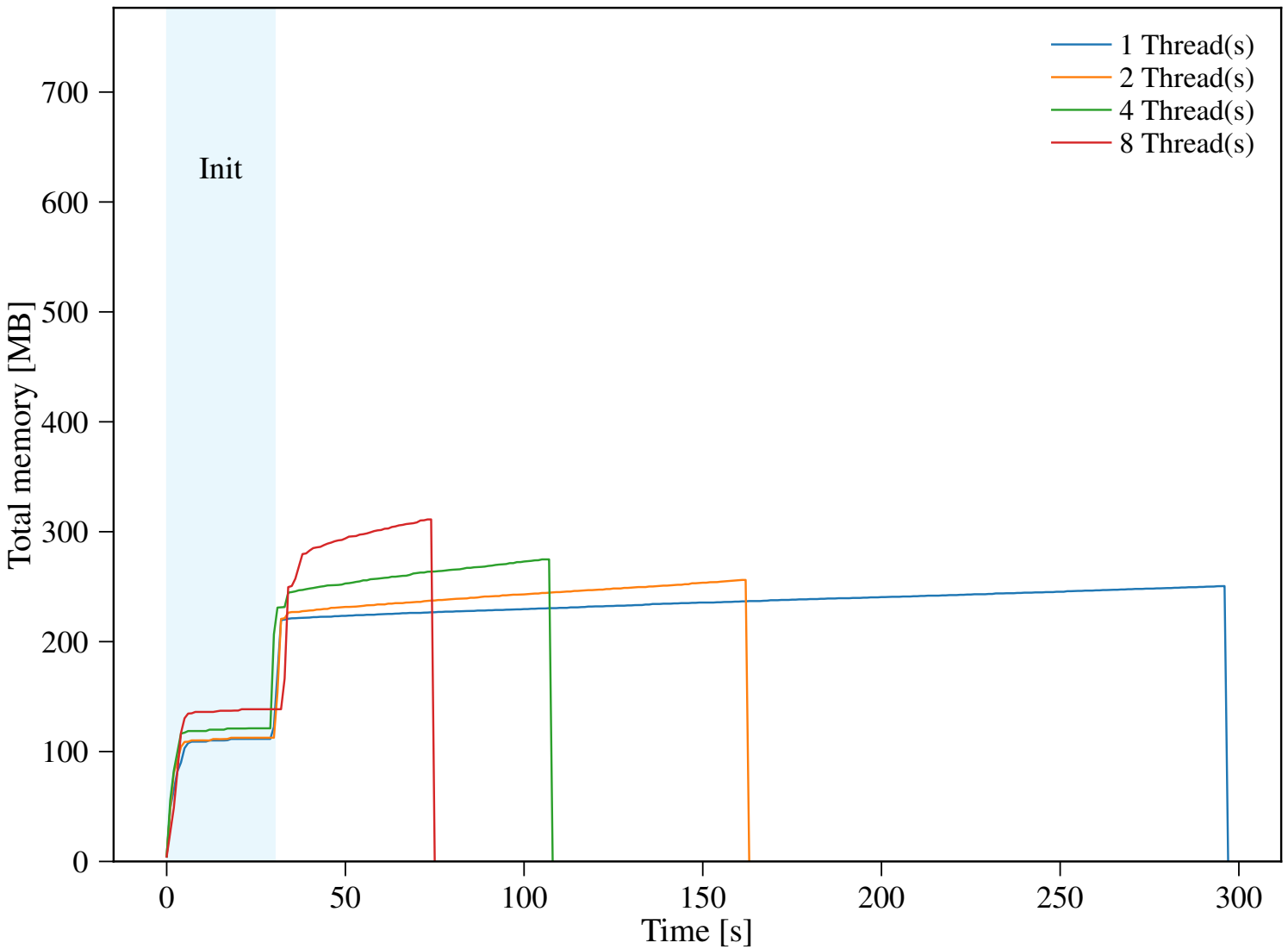
Memory scaling



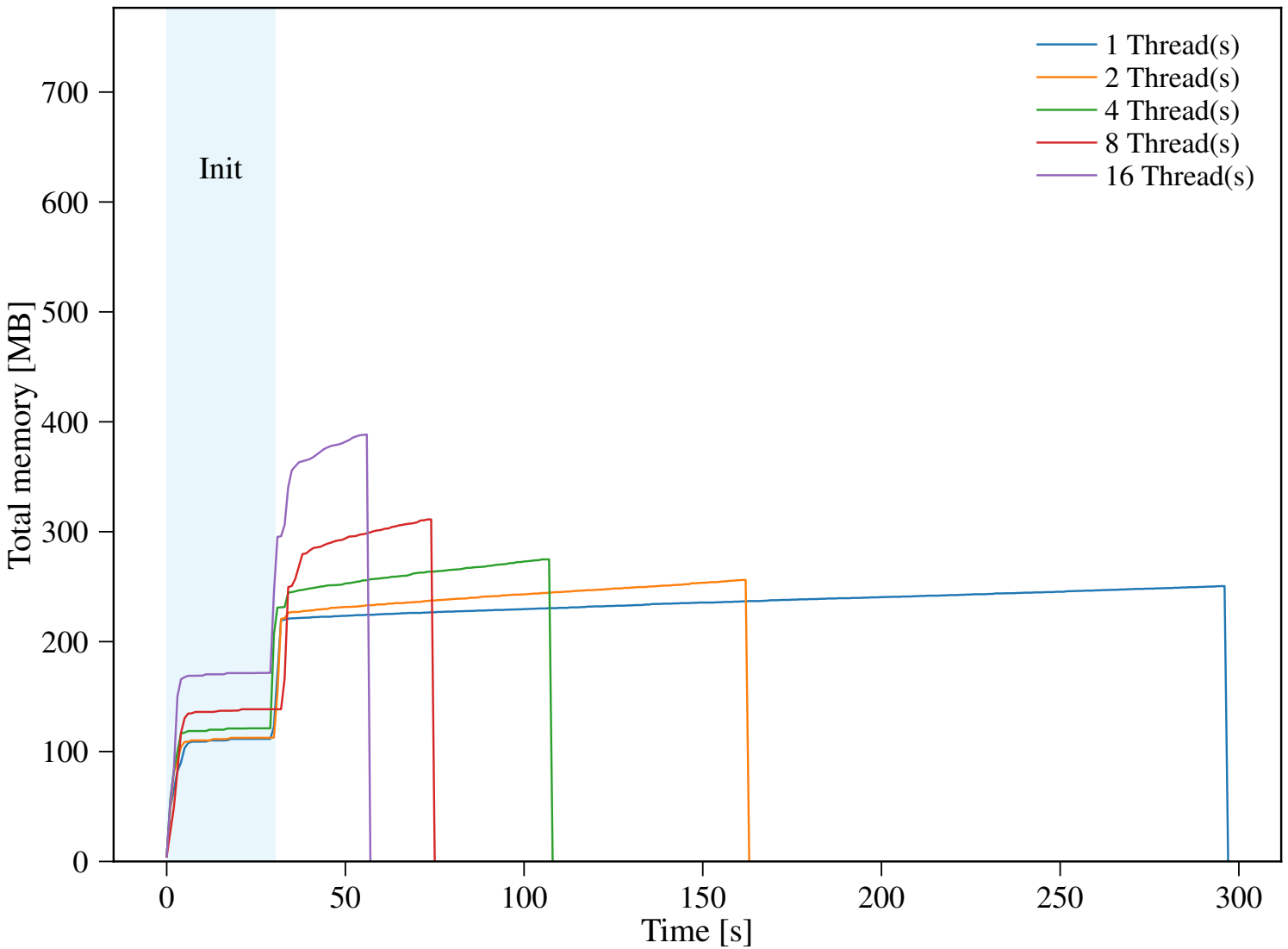
Memory scaling



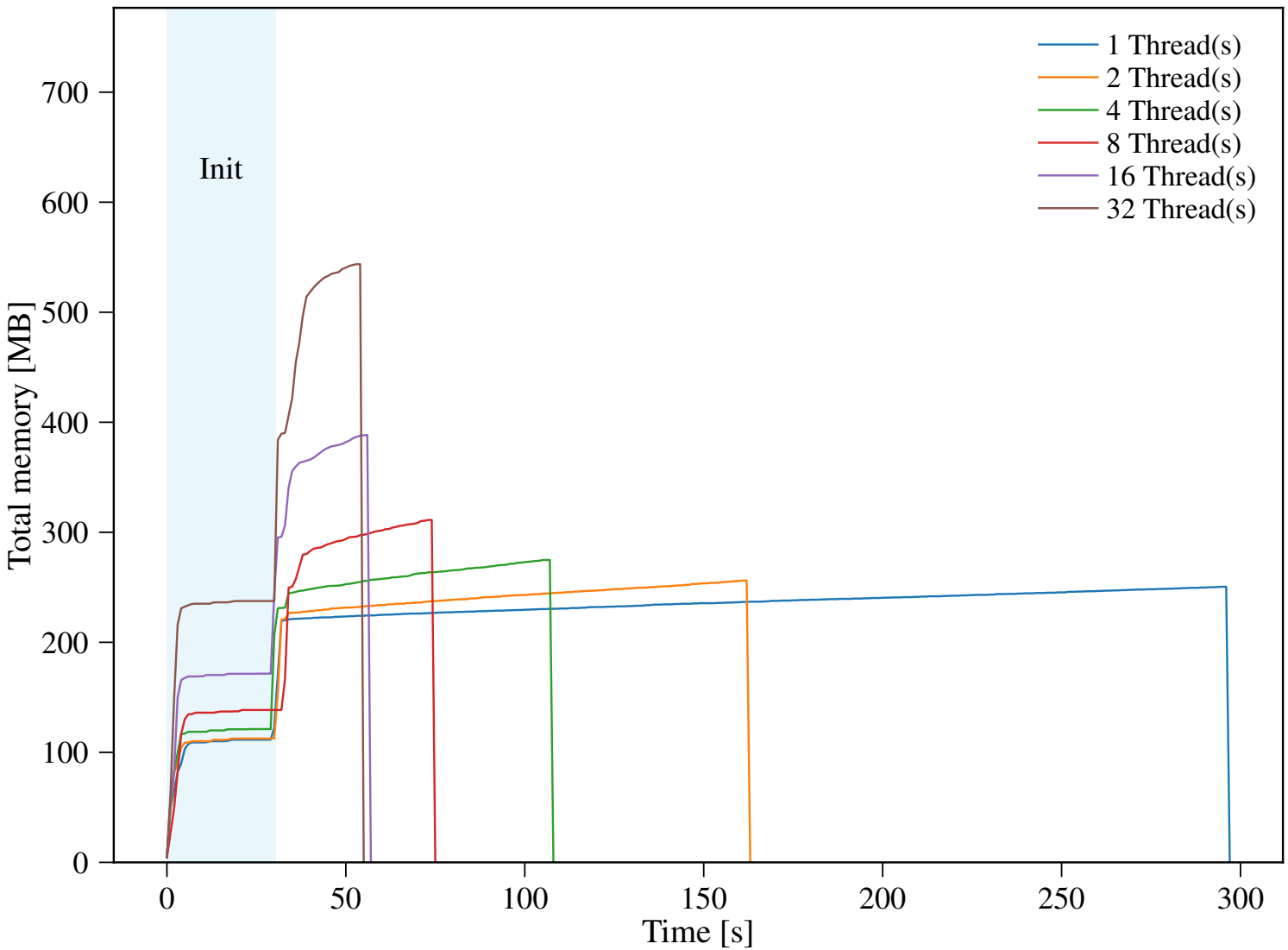
Memory scaling



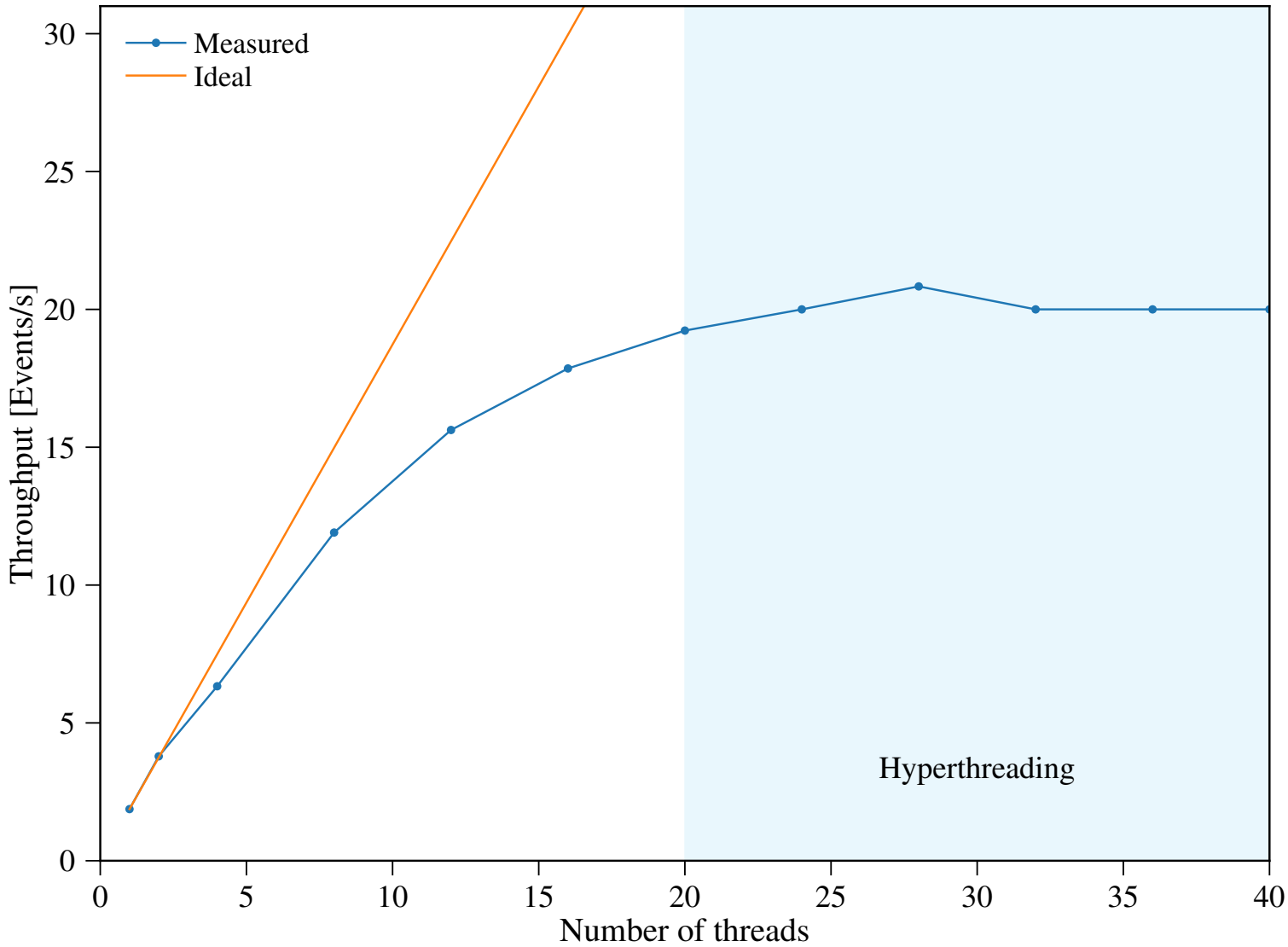
Memory scaling



Memory scaling



Throughput


Measured
Ideal
Hyperthreading
Throughput [Events/s]
Number of threads
0
5
10
15
20
25
30
35
40

Summary

- Gaussino, an LHCb-independent core framework, well advanced.
  - Modular generation phase with Pythia8 example
  - Prototype for the interaction with Geant4 MT
  - Encouraging first results

Outlook

- Test performance with a complete detector
- Migrate Gauss to be based on Gaussino